Michigan Outdoor Recreation Search Interest Installment 7

This is my project to analyze and forecast Google search interest for 10 forms of oudoor recreation by people in Michigan. My data is daily data for each form of search interest from January 2021 to March 2025. The search interest terms for this project are atving, boating, camping, fishing, hiking, kayaking, rving, hunting, skiing, and snowmobiling. For more information on how this data is pulled, please see installment 5:

https://dataandoutdoors.com/michigan-outdoor-recreation-search-interest-installment-5/

This installment is my second of two or three attempts/methods to model and forecast search interest based on this data.

Nonstationary Model

The nonstationary model is the same as for Installment 6. I employ a hankel matrix with 366 rows. I denoise the data using singular value decomposition. Then I predict the last row of data using the previous 365 using (essentially) OLS estimation. This process works for all ten data series except for camping, which will be ommitted again for this installment. For more details, please see Installment 6.

https://dataandoutdoors.com/michigan-outdoor-recreation-search-interest-installment-6/

Stationary Model

Once I estimate the stationary models above, I extract the 1,185 residuals from the estimation of each of the 10 series. These residuals are considered to be stationary. Unlike the setup I learned in the class project where we estimated a single stationary series and employed an ARIMA model, here I combine all ten series. In the previous installment I used a Vector AutoRegressive (VAR Model). The VAR model considers not only lagged values of the same series (say ATVing) but also lagged values of other series (such as camping). I considered up to one lag of each series. However, the disadvantage of the VAR model is it's somewhat difficult to incorporate nonlinear features and interactions. For instance, it's possible that the impact of weather is different for different seasons or different holidays. Further, different levels of weather might have different impacts. Therefore, for this installment, I use a feed forward neural network instead of a VAR model.

As with the previous installment, I consider 'exogenous' explanatory variables. These are four daily weather variables: precipitation, snow depth, maximum temperature, minimum temperature. These series are the average of the values taken from 105 different weather stations throughout Michigan. For more information on how these data series were created, please see installment 3:

https://dataandoutdoors.com/michigan-outdoor-recreation-search-interest-installment-iii/

Additionally, for this installment, I add variables for calendar day (1-365), day of week (0-6), and the week of (within three days) of several major holidays: New Year's Day, Valentine's Day, Memorial Day, Independence Day, Labor Day, November 15th, Thanksgiving, Christmas Day, and New Year's Day. November 15th is included for hunting. Valentine's day is included because it's during ski season and likely a popular holiday for skiing trips. Note that all of these additional time variables are related to seasonality, but seasonality in each series is already accounted for in the non-stationary model. Nonetheless, this will account for the interaction between weather variables and different times of year, week, and holidays. Also, the previous models don't account for the impact of holidays, or their interaction with holiday weekends.

For all previously included variables, please see discussion in Installment 6.

https://dataandoutdoors.com/michigan-outdoor-recreation-search-interest-installment-6/

The aspects of the feed foward neural network are commonly known and can easily be encountered online if they are not well known to you. These neural networks are comprised by layers of nodes. The nodes of the first layer feed into the next layer and so on. Each of the nodes make a nonlinear transformation of the underlying linear index functions, i.e. linear functions of the model variables. The output of the nodes of one layer are passed onto all the nodes in the next layer. Given that the neural network combines many nonlinear transformations (multiple nodes) multiple times (multiple layers) they are able to fit very complex nonlinear transformations without explicitly specifying them.

As with the VAR model, the neural network is capable of fitting a single model for all of the nine output variables. However, as I did with the VAR models, I choose to fit a completely separate model for each of the outputs. The reason is that the relationship between the underlying variables is likely very different, as it was with the VAR model. For each model I consider two hyperparameters:

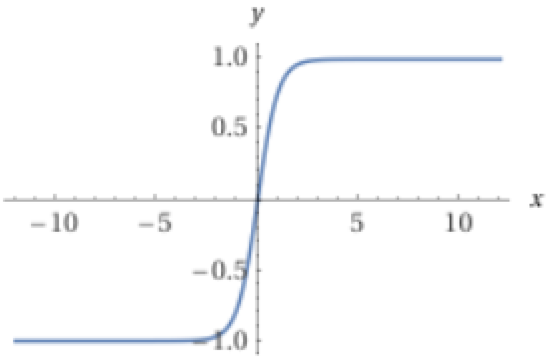
1.) Number of Layers: This is the number of layers of nodes. I compare the results for 1, 2, & 3 layers.

2.) Number of Nodes: This is the number of nodes per layer. I test 1-5. To reduce the dimensionality of my testing, I require the same number of nodes per layer.

Since I require the number of nodes to be the same for each layer, there are 15 different models I test (for each of the nine series).

For each of the 15 models, I must choose an activation function or the nonlinear transformation of each of the nodes. I set the activation function deterministically based on the number of layers. For one or two layers, I use the hyperbolic tangent function. For three layers, I use the RELU function.

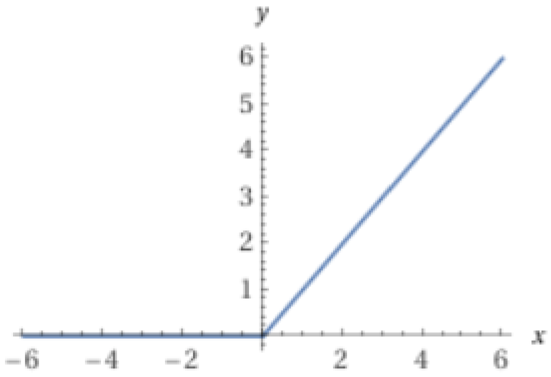
In the picture below, I show the hyperbolic tangent function which was graphed with wolfram alpha. Note that this function looks very similar to the sigmoid/logistic regression function. The marginal change (derivative) is highest in the middle and as 'x' becomes infinitely high or low then the value of the fuction limits/assymptotes to a value. However, unlike the logistic regression function that assymptotes to 0 an 1, the hyperbolic tangent assymptotes to -1 and 1. This makes it easier for the neural network to converge.



(x from −12 to 12)

However, I don't use the hyperbolic tangent function with three layers. The hyperbolic tangent function can struggle to converge with many layers due to an issue known as 'vanishing gradient'. To understand this issue, recall that the neural network uses the first derivative of the neural network to converge on a solution. Note that for high or low values the first derivative of the hyperbolic tangent is near zero. Hence, it's difficult for the neural network algorithm to determine whether an incremental change in model parameters improves the fit of the model. That is especially when multiple gradients (layers) are multiplied together. A single node being numerically near zero will cause the entire gradient to be zero and no update in the algorithm.

Instead of the hyperbolic tangent, I use the ReLU function for three layers. This rather simple piecewise function is zero when 'x' is negative and the identity function when 'x' is possible. This piecewise transformation of a linear function creates a nonlinear function for the purpose of the neural network which is also resistant to the issue of vanishing gradient.



(x from −6 to 6)

I choose my hyperparameters (nodes and layers) based on cross validation. That is because neural networks do not have standardized statistical tests such as the VAR model which I could use to specify the model. The usual method of performing cross validation is to choose a random hold out dataset that the model doesn't see in training to determine the prediction success of each choice of hyperparameters.

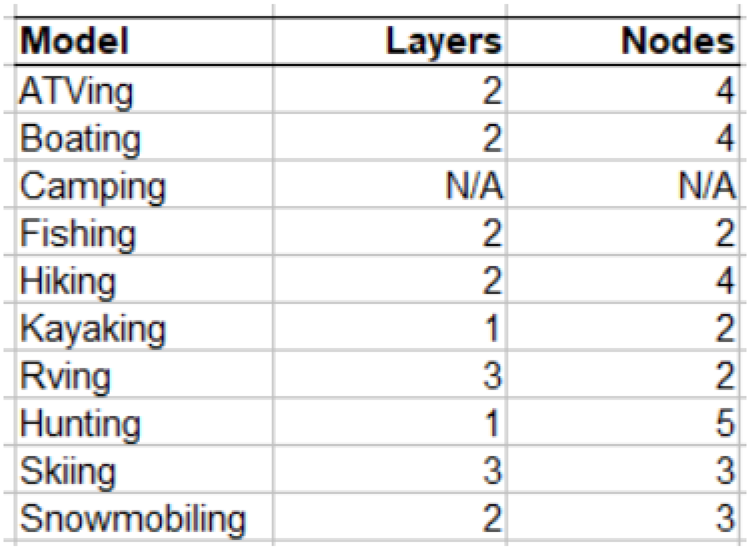
However, in the case of time series data it's acceptable to only choose validation data that occurs after the data used to train the model.

Therefore, I train each of the 15 possible versions of the model using data from January 2022 to March 2024. The final year of data (April 2024-March 2025) is used as validation data. Since the models will be used to provided 14 day forecasts, it is not prudent to make a single 365 day forecast. (Indeed, the models would probably not perform very well forecasting months ahead of time.) Instead, I create 352 14 day forecasts starting with each day of the year and average the Mean Squared Error throughout the days.

Potential disadvantages of this approach is that the cross validation is only based on performance of the last year. However, I wanted sufficient data to train the model (i.e. at least two years of data). This length of time is known as burn-in. Another disadvantage is that I only trained the models once when, in theory, I could have trained the model with more data each time I increment forward by one day with the forecast window. I've personally done this on the Northern Michigan Search Interest project. However, in this case I compromised my approach due to the computational feasibility of training 15 neural networks hundreds of times.

A final disadvantage is that model errors are averaged over the entire year which may not consider when model forecasts are more important. For instance, search interest for skiing is most likely low except for a few months a year. If a model simply forecasted the same low value all year, it might achieve a relatively low MSE. Therefore, for skiing and snowmobiling, I use only winter months for cross validation. For hunting, I use fall months. For fishing, ATVing, and RVing, I use the entire year. For all other categories, I use performance during summer only to choose the nodes and layers.

In order to choose the number of layers, I average the MSE for the five choices of nodes for each number of layers. I then choose the number of layers with the lowest average MSE. Once I choose the number of layers, I then choose the number of nodes with the lowest MSE within that number of layers. In the chart below, I show the number of layers and nodes chosen for each of the models.



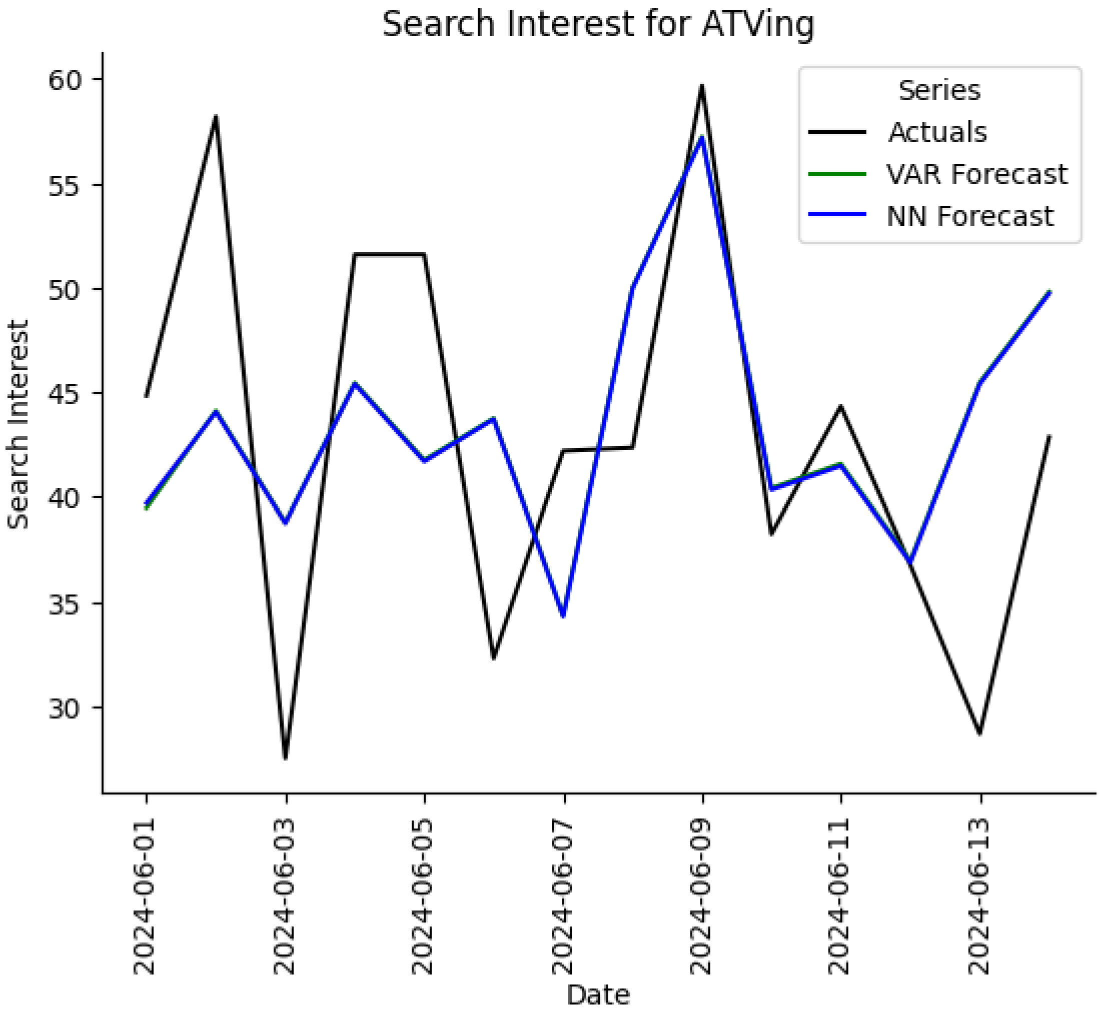
Model	Layers	Nodes
ATVing	2	4
Boating	2	4
Camping	N/A	N/A
Fishing	2	2
Hiking	2	4
Kayaking	1	2
Rving	3	2
Hunting	1	5
Skiing	3	3
Snowmobiling	2	3

Selected Forecasts

Just as in Installment 6, I choose a different forecast period for each of the data series. In addition to plotting actuals and the results of Model 1, I now overlay the results from Model 2. Please see Installment 6 for more details on these forecast periods. Note, that this isn't intended to be a comprehensive evaluation of how the models will perform over various time periods. Instead, these forecasts are intended for illustrative purposes only.

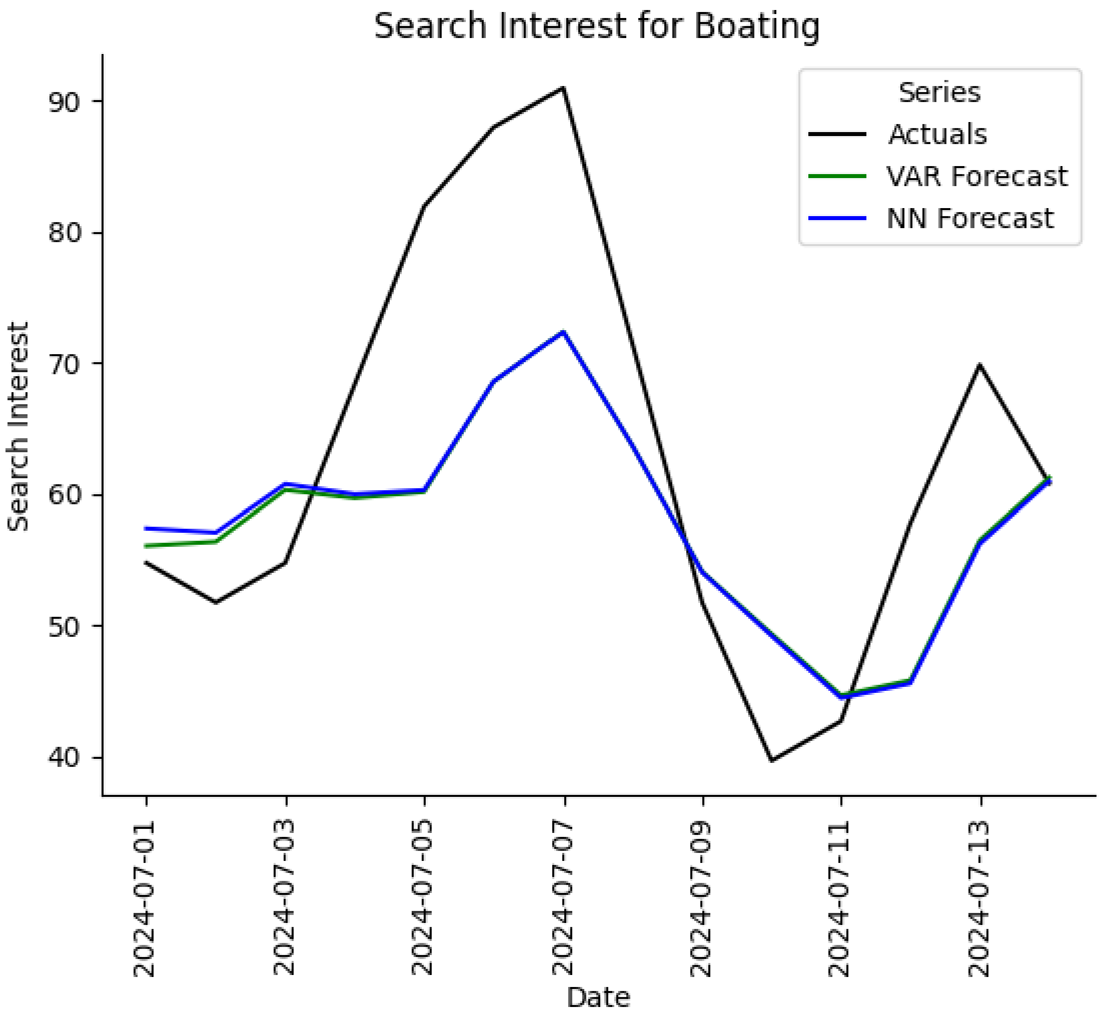
ATVing

The neural network results are almost exactly the same as the VAR model results. See Installment 6 for more information.



Boating

The neural network results are almost exactly the same as the VAR model results. That's despite having variables for the fourth of July weekend. Please see Installment 6 for more discussion of these forecasts.

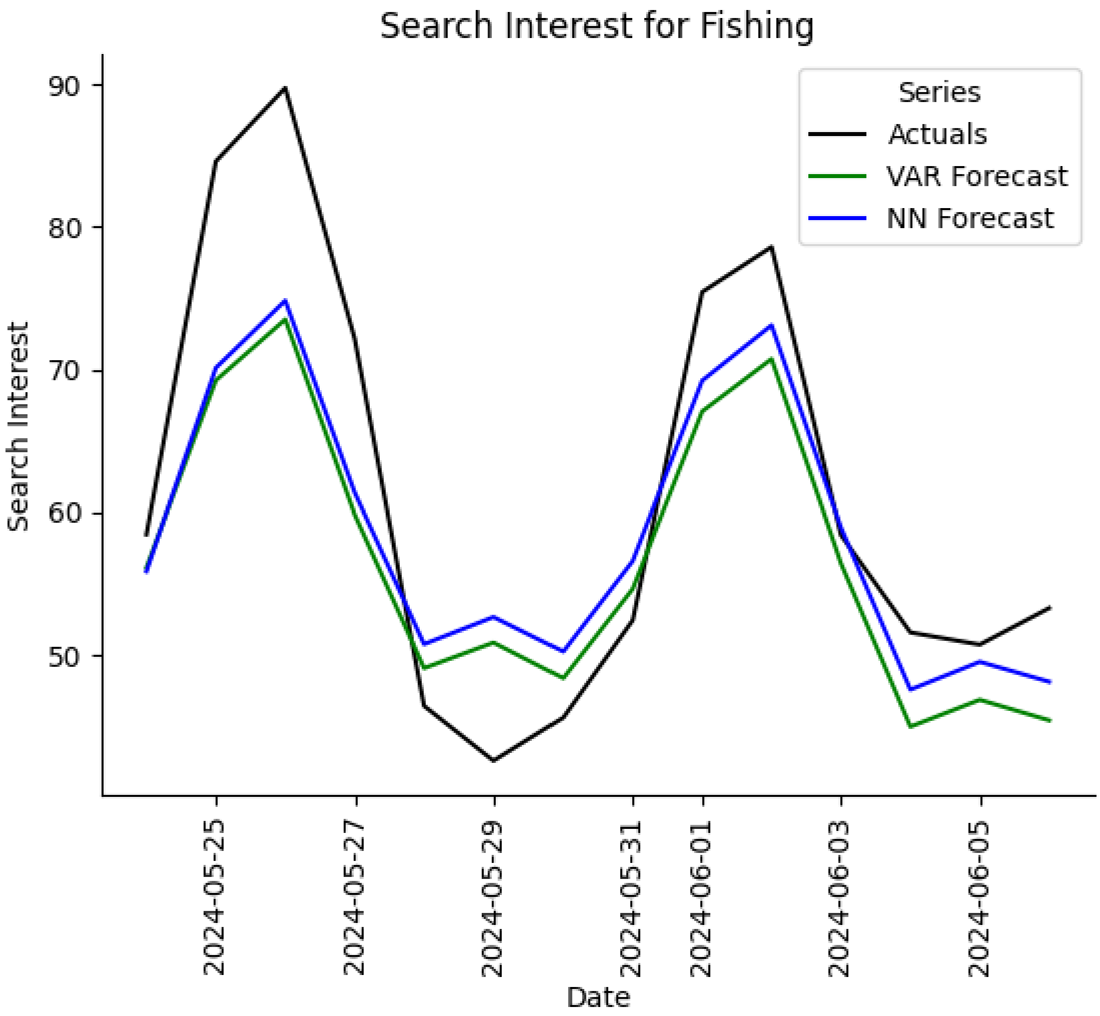


Camping

Unfortunately, for camping I wasn't able to produce a forecast with this methodology. As can be seen above, the hankel matrix for camping only has 12 components compared to at least 50 for the other series. This multicolinearity in the values has made it difficult to fit the nonstationary model. The model for camping will have to remain an area of future research. This same issue impacts both the VAR and neural network models.

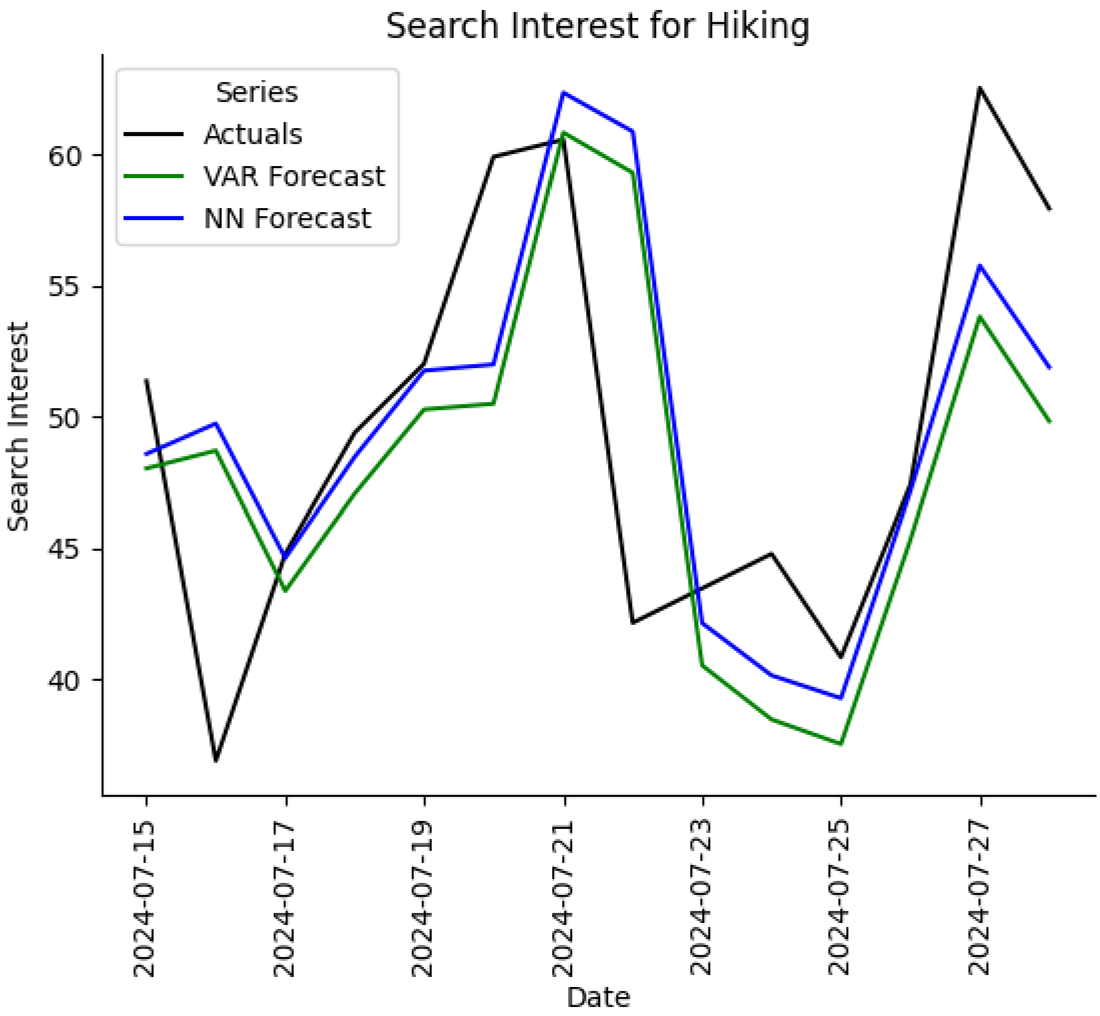
Fishing

The neural network forecast for fishing is slightly higher and closer to actuals than the VAR model. One would think that inclusion of holiday variables would make the forecast closer than it is.



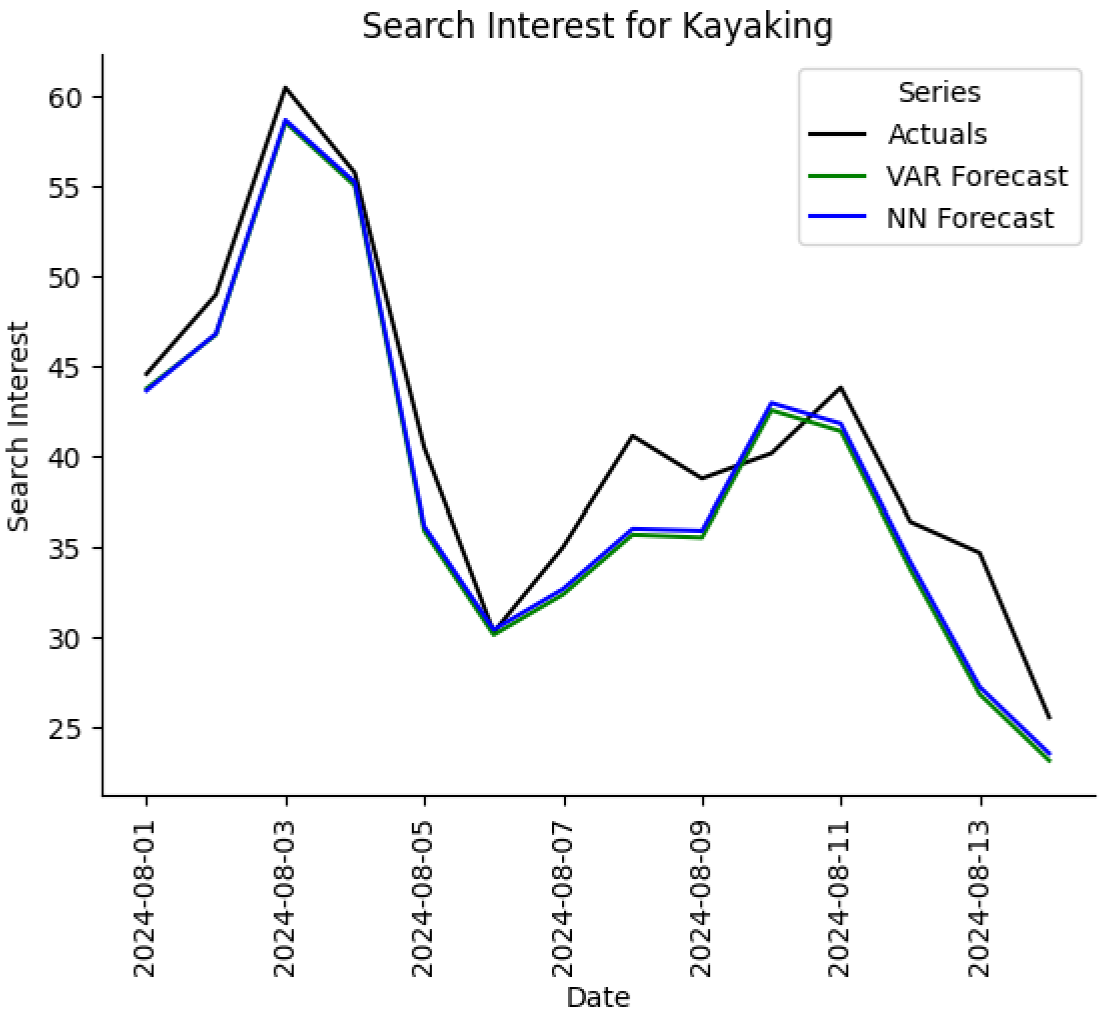
Hiking

For the forecast starting July 15, the neural network forecast is higher than the VAR forecast and, over a majority of the time interval, closer to actuals.



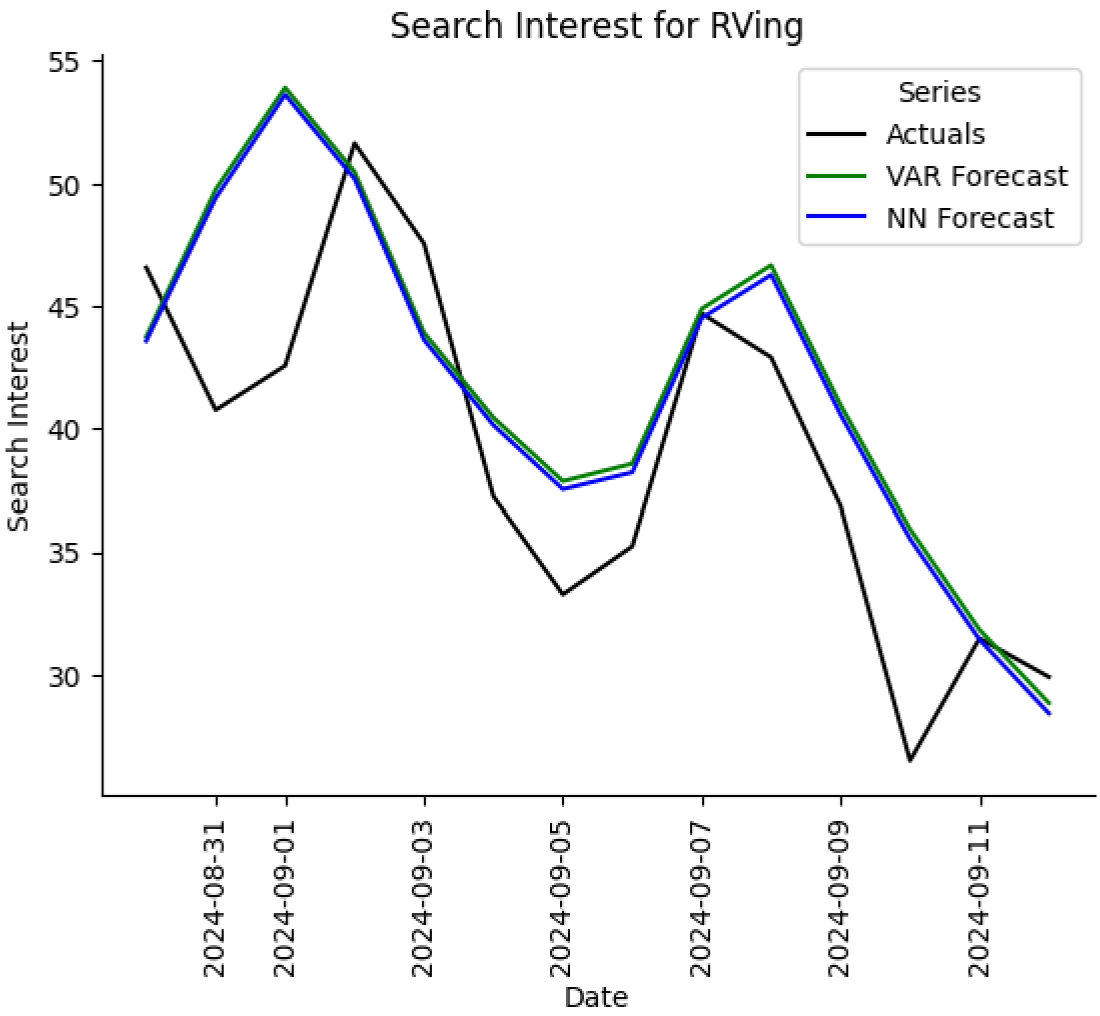
Kayaking

I provide a forecast for Kayaking starting August 1. The neural network and the VAR model have very similar forecasts, both of which track actuals very well.



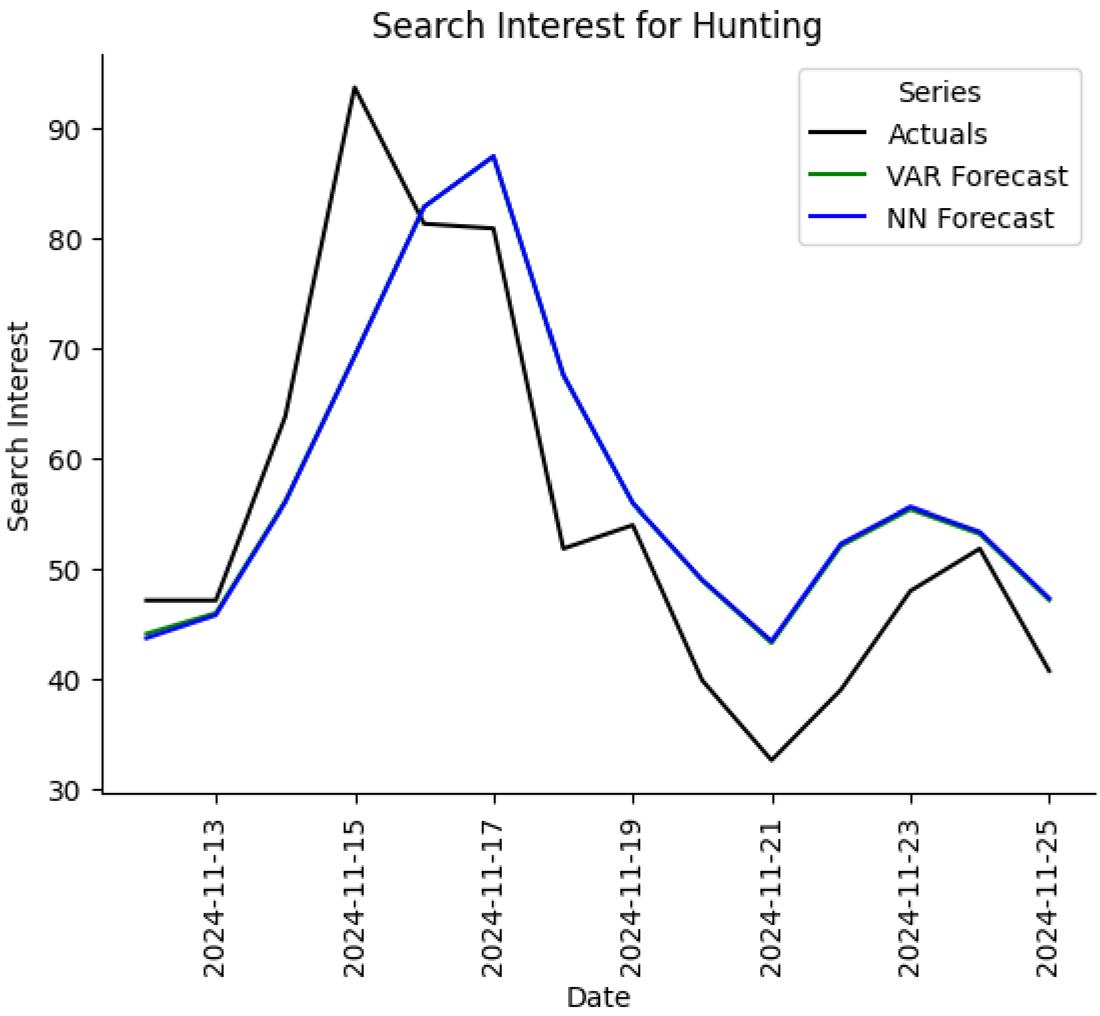
RVing

My forecast for RVing is over labor day weekend and the neural network and VAR model forecassts are very similar. Both imperfectly account for holidy weekends, despite the neural network containing variables for holidays. Please see Installment 6 for more discussion of this forecast.



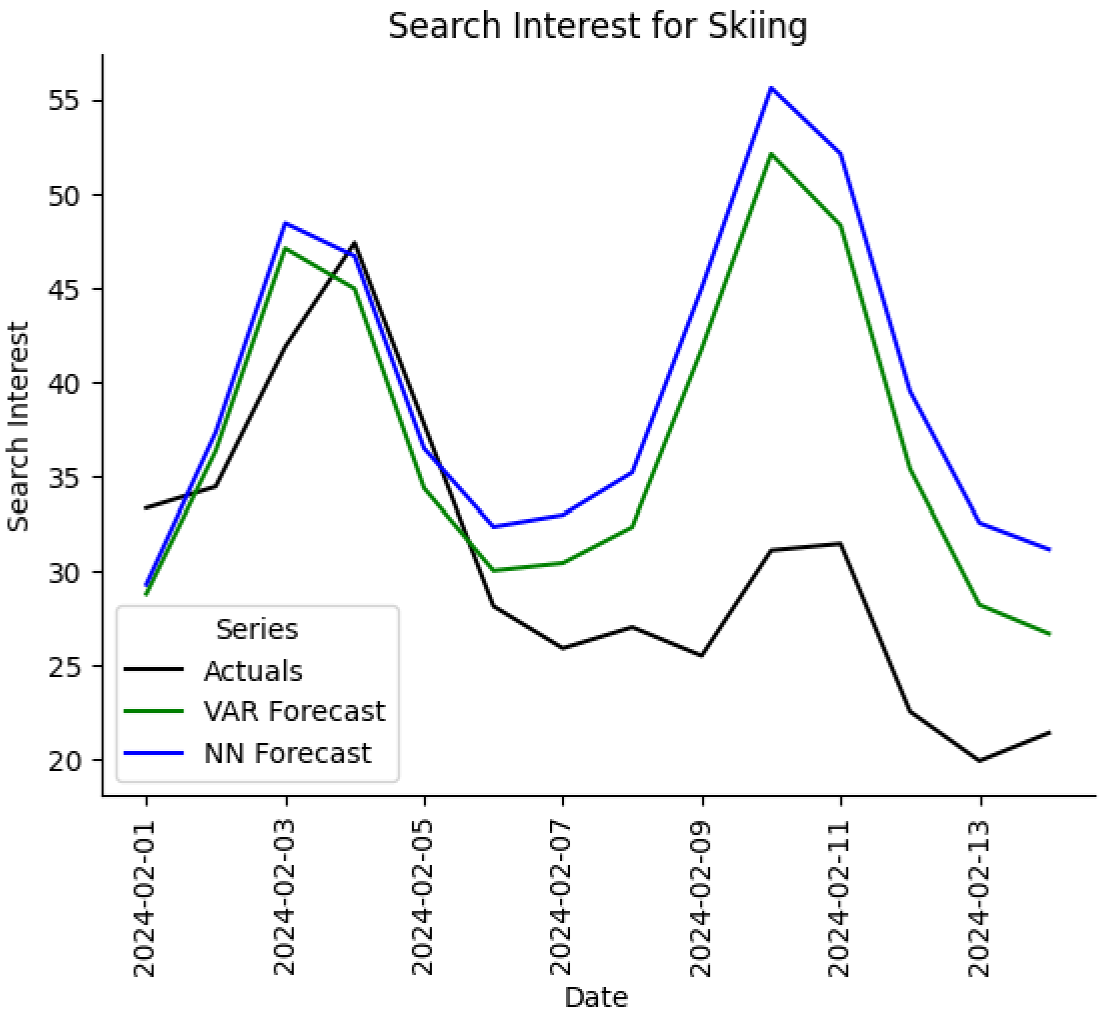
Hunting

I provide a forecast for hunting over November 15th, which is the opening of firearms deer season in Michigan. The forecast is close to actuals, but there is little difference between the neural network and VAR forecast. Please see Installment 6 for more discussion of this forecast.



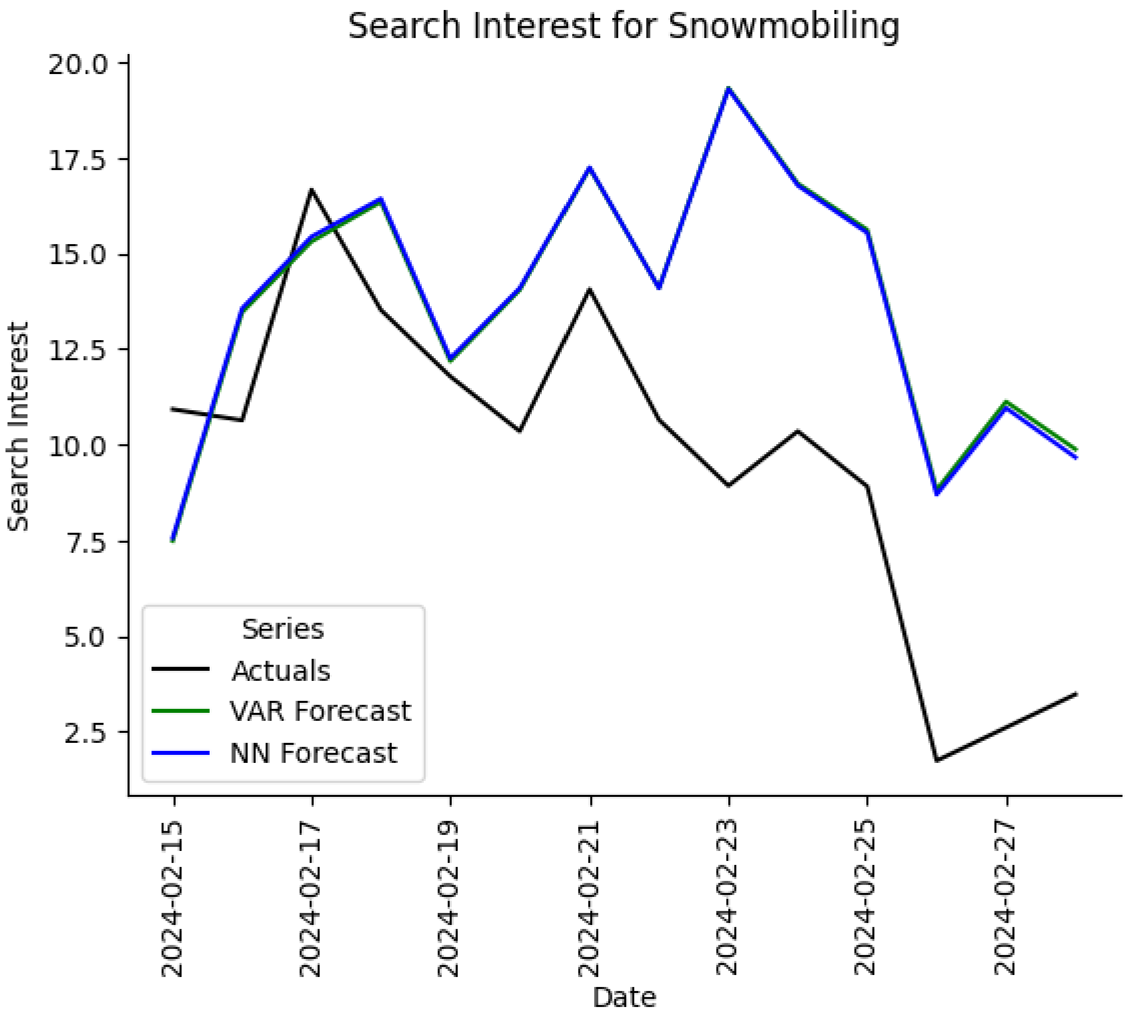
Skiing

I provide a forecast for skiing starting February 1st. The neural network is higher and further from actuals than the VAR model. Unfortunately, the neural network doesn't appear to capture the impact of poor snowfall in 2024.



Snowmobiling

I provide a forecast for snowmobiling starting on February 15th. The neural network and the VAR forecast are almost exactly the same and neither capture the impact of poor snowfall in 2024.



Conclusion

This is my second model option for forecasting search interest for ten forms of outdoor recreation in Michigan. Results for the neural network are dissapointingly close to the VAR model forecasts. In most cases, the nonstationary model dominates the forecast. Despite providing additional variables to the neural network for holidays and time of year, the neural network doesn't capture holidays or the impact of snowfall much better than the VAR model.

This result is less than hoped but not astoundingly unexpected. There are only a few hundred observations in the dataset and, despite the theoretical ability of a neural network to fit complex nonlinear features, truth is they aren't magic.

I have one additional modeling technique planned. In the end, the first two methods have done ok. Once I have one more option, I will likely improve the best option. For instance, for the VAR model I could manually add features for holidays or interactions between weather and specific seasons. In the case of the neural network, I could reduce the variables in the model so it isn't trying to fit too many irrelevant features.